Research Heaven,
West Virginia

See More, Learn More, Tell More

(briefing)

Tim Menzies
West Virginia University
tim@timmenzies.net

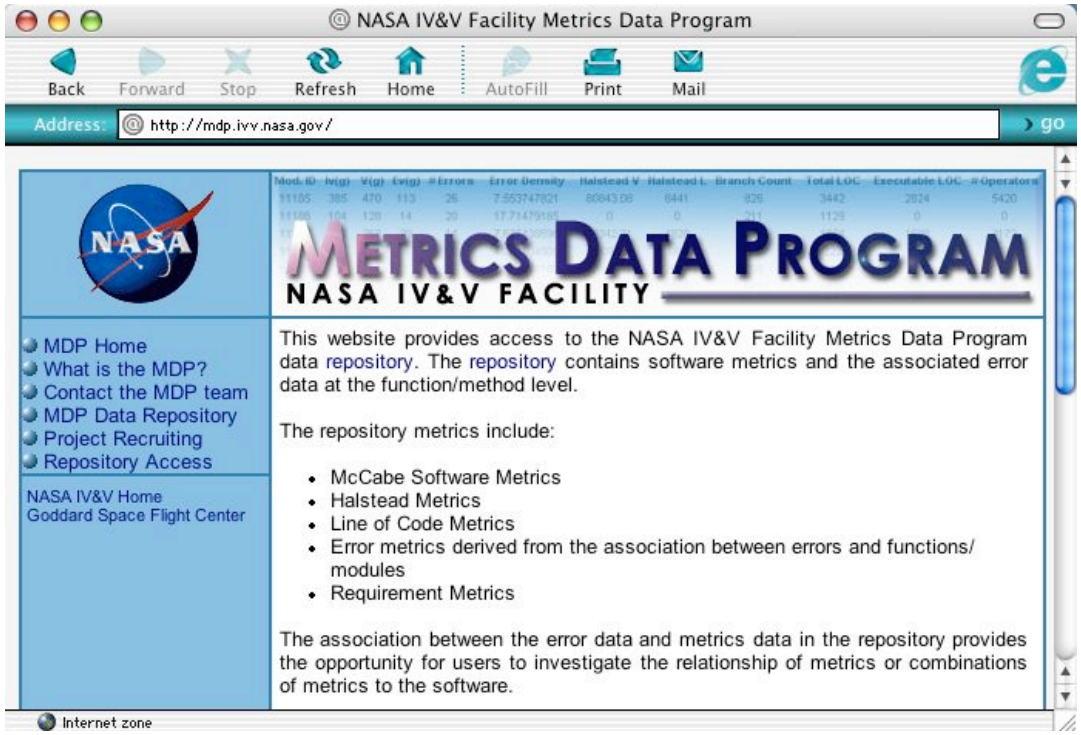
Problem

- **Mountains of data**
  - Seek “the diamonds in the dust”
- **We have many do-ings**
  - But what are we learn-ing?
- **What general lessons about software quality assurance can we offer NASA?**
- **Problem of external validity**
  - It worked “there” but will it work “here”?

Approach

Research Heaven,
West Virginia

- **while not ((end of time OR end of money))**
  - chase data sets
  - extract cost-benefit patterns from data
  - check the stability of those patterns
  - report stable conclusions
- **Product metrics:**
  - NASA metric's data program
  - Goddard project
  - Flight simulators
- **Process metrics:**
  - cost estimation data from JPL
    - Now spun off into a project with Jairus Hihn
  - SILAP (IV&V effort potential model)



http://mdp.ivv.nasa.gov

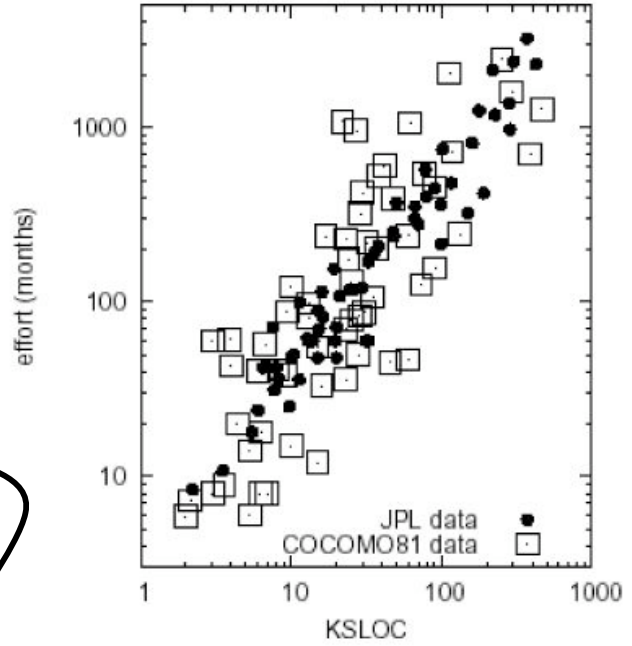
Now with 10+ projects ⇐ and many more soon

Importance/ Benefits

- **Generally:**
  - –NASA does a lot of software
  - –What guidance should we offer developers?
  - –How good is that guidance
    - Has that guidance been certified?
    - Do we know how general are those guidelines?

Relevance To NASA

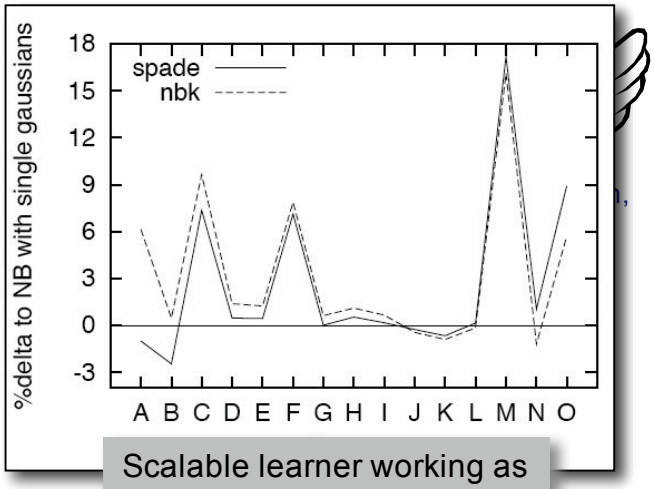
- **Data comes from NASA**
  - Process metrics:
    - JPL project data
    - IV&V effort potential data
  - Product metrics
    - Defect logs from multiple NASA centers
    - Flight simulator data
- **Conclusions apply to NASA projects**



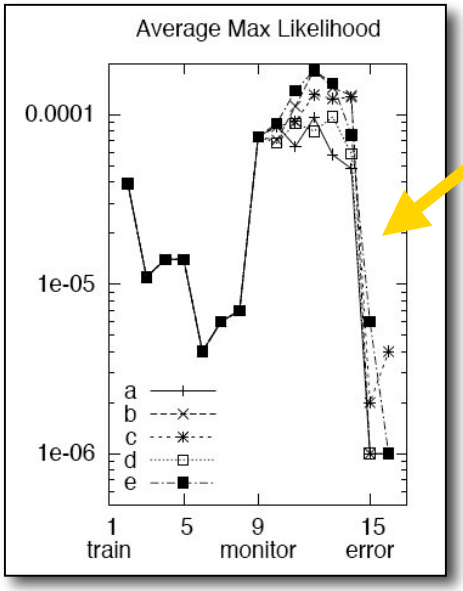
project	# modules	% with defects	language	developed at	notes
CM1	496	9.7%	C	location 2	a NASA spacecraft instrument
JM1	10885	19%	C	location 3	real-time predictive ground system: uses simulations to generate the predictions
KC1	2107	15.4%	C++	location 4	storage management for receiving and processing ground data
KC2	523	20%	C++	location 4	science data processing; another part of the same project as KC1; different personnel to KC1. shared some third-party software libraries as KC1, but no other software overlap.
PC1	1107	6.8	C	location 5	support tools
Total	15118				

Accomplishments

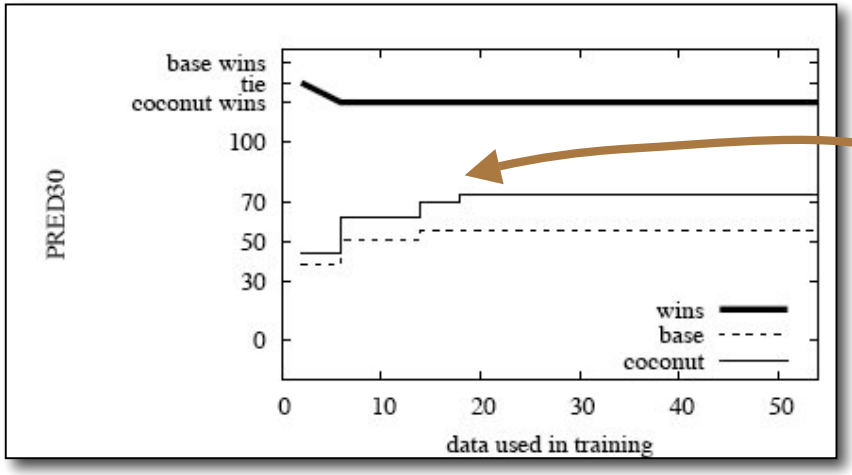
- **Before:**
  - –Can automatically learn defect detectors from error logs
  - –Those defect detectors from code are much BETTER than previously believed
    - Yes, false negative, but adequate to good detection probabilities
    - (Enough) stability across multiple projects
- **Now:**
  - –Can automatically learn software cost models
    - AND determine how much data is required to do that
  - –Can scale up to HUGE data sets
  - –Can determine when a learned theory goes “out of scope”
  - –See also “SPOT/CUBE” in “martha”



Scalable learner working as well as state-of-the-art, non-scalable, alternative



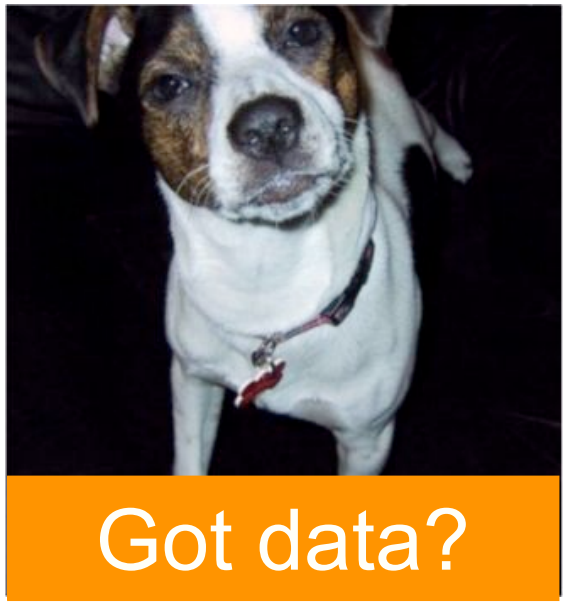
Where a learner has left the zone where it was certified



When we have seen enough data to learn a good cost model

Next Steps

Research Heaven,
West Virginia



- **Data mining needs data**
  - Got data?
    - Then meet your new best friend
- **Current plans**
  - More defect data studies
    - Dozens, not just 5, data sets
    - Check effectiveness and stability?
  - Release of the generalized toolkits
    - Tutorials
    - manuals
  - Generalized anomaly detectors
    - The “selection bias” problem
  - Synergies with other SARP data mining projects